One-way quantum finite automata together with classical states*

Daowen Qiu[a,b,c,†], Paulo Mateus[b,‡], Xiangfu Zou[a], Amilcar Sernadas[b,‡]

[a] *Department of Computer Science, Zhongshan University, Guangzhou 510275, China*

[b] *SQIG–Instituto de Telecomunicações, IST, TULisbon,*
Av. Rovisco Pais 1049-001, Lisbon, Portugal

[c] *The State Key Laboratory of Computer Science, Institute of Software,*
Chinese Academy of Sciences, Beijing 100080, China

Abstract

One-way quantum finite automata (1QFA) proposed by Moore and Crutchfield and by Kondacs and Watrous accept only subsets of regular languages with bounded error. In this paper, we develop a new computing model of 1QFA, namely, *one-way quantum finite automata together with classical states* (1QFAC for short). In this model, a component of classical states is added, and the choice of unitary evolution of quantum states at each step is closely related to the current classical state. 1QFAC can accept all regular languages with no error, and in particular, 1QFAC can accept some languages with essentially less number of states than *deterministic finite automata* (DFA). The main technical results are as follows. (1) We prove that the set of languages accepted by 1QFAC with bounded error consists precisely of all regular languages. (2) We show that, for any prime number $m \geq 2$, there exists a regular language $L^0(m)$ whose minimal DFA needs $O(m)$ states and that can not be accepted by the 1QFA proposed by Moore and Crutchfield and by Kondacs and Watrous, but there exists 1QFAC accepting $L^0(m)$ with only constant classical states and $O(\log(m))$ quantum basis states. Analogous results for multi-letter automata are also established. (3) By a bilinearization technique we prove that any two 1QFAC $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent if and only if they are $(k_1n_1)^2 + (k_2n_2)^2 - 1$-equivalent, and there exists a polynomial-time $O([(k_1n_1)^2 + (k_2n_2)^2]^4)$ algorithm for determining their equivalence, where k_1 and k_2 are the

*This work is supported in part by the National Natural Science Foundation (Nos. 60573006, 60873055), the Research Foundation for the Doctorial Program of Higher School of Ministry of Education (No. 20050558015), Program for New Century Excellent Talents in University (NCET) of China, and by project of SQIG at IT, funded by FCT and EU FEDER projects Quantlog POCI/MAT/55796/2004 and QSec PTDC/EIA/67661/2006, IT Project QuantTel, NoE Euro-NF, and the SQIG LAP initiative.

†Corresponding author. *E-mail address:* issqdw@mail.sysu.edu.cn (D. Qiu).

‡*E-mail address:* {paulo.mateus,amilcar.sernadas}@math.ist.utl.pt.

numbers of classical states of $\mathcal{A}_1$ and $\mathcal{A}_2$, as well as n_1 and n_2 are the numbers of quantum basis states of $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. Finally, some other issues are addressed for further consideration.

Keywords: Quantum computing; Quantum finite automata; Equivalence; Regular languages; Deterministic finite automata; Decidability; State complexity; Topological automata

1. Introduction

Quantum computers—the physical devices complying with quantum mechanics were first suggested by Benioff [7] and Feynman [12] and then formalized further by Deutsch [11]. A main goal for exploring this kind of model of computation is to clarify whether computing models built on quantum physics can surpass classical ones in essence. Actually, in 1990's Shor's quantum algorithm for factoring integers in polynomial time [34] and afterwards Grover's algorithm of searching in database of size n with only $O(\sqrt{n})$ accesses [13] have successfully shown the great power of quantum computers. This intriguing field has attracted much attention since then [25, 14]. Clarifying the power of some fundamental models of quantum computation has become of interest [14].

Quantum finite automata (QFA) can be thought of as a theoretical model of quantum computers in which the memory is finite and described by a finite-dimensional state space [1], as finite automata are a natural model for classical computing with finite memory [17]. As mentioned in [16], one of the motivations to study QFA is to provide some ideas to investigate the relation of classical and quantum computational complexity classes. This kind of theoretical models was firstly studied by Moore and Crutchfield [23], Kondacs and Watrous [19], and then Ambainis and Freivalds [2], Brodsky and Pippenger [9], and other authors (e.g., the references in [29]). The study of QFA is mainly divided into two ways: one is *one-way quantum finite automata* (1QFA) whose tape heads move one cell only to right at each evolution, and the other *two-way quantum finite automata* (2QFA), in which the tape heads are allowed to move towards right or left, or to be stationary. According to the measurement times in a computation, 1QFA have two types: *measure-once* 1QFA (MO-1QFA) initiated by Moore and Crutchfield [23] and *measure-many* 1QFA (MM-1QFA) studied first by Kondacs and Watrous [19]. In MO-1QFA, there is only a measurement for computing each input string, performing after reading the last symbol; in contrast, in MM-1QFA, measurement is performed after reading each symbol, instead of only the last symbol.

Though MM-1QFA can accept more languages than MO-1QFA with bounded error [2], both of them accept proper subsets of regular languages [9, 8]. Another model of 1QFA with a measurement is called *multi-letter* 1QFA, proposed in [6]. In multi-letter 1QFA,

there are multi-reading heads. Roughly speaking, a k-letter 1QFA is not limited to seeing only one, the just-incoming input letter, but can see several earlier received letters as well. Though multi-letter 1QFA can accept some regular languages not acceptable by MM-1QFA, they still accept a proper subset of regular languages. Therefore, in general, 1QFA with a measurement accept some proper subsets of regular languages, but some models of 1QFA with measurement performed after reading each input symbol can accept all regular languages; for example, 1QFA *with control languages* (1QFACL, for short) proposed in [8] accept all regular languages (and only regular languages) [24]. However, the measurements in 1QFACL differ from those in MM-1QFA proposed in [19].

Paschen [26] presented a different 1QFA by adding some ancilla qubits to avoid the restriction of unitarity, and this model is called an ancilla 1QFA. Indeed, in ancilla 1QFA, the transition function corresponding to every input symbol is described by an isometry mapping, instead of a unitary operator. In [26], it was proved that ancilla 1QFA can recognize any regular language with certainty. Following the idea in Bennett [4], Ciamarra [10] proposed another model of 1QFA whose computational power was shown to be at least equal to that of classical automata. For convenience, we call the 1QFA defined in [10] as *Ciamarra 1QFA* named after the author. In fact, the internal state of a Ciamarra 1QFA evolves by a trace-preserving quantum operation. In addition, in [22] it was proved that both ancilla 1QFA and Ciamarra 1QFA recognize only regular languages.

Hence, proposing and exploring some more practical models of quantum computation is an intriguing problem and may provide more help to study the physical models of quantum computers. Indeed, motivated by the implementations of quantum computers using nucleo-magnetic resonance (NMR), Ambainis et al. [1] proposed another model of 1QFA, namely, *Latvian* 1QFA (L-1QFA, for short). In L-1QFA, measurement is also allowed after reading each input symbol, but they also accept a proper subset of regular languages [1].

Though ancilla 1QFA and Ciamarra 1QFA can accept all regular languages, their evolution operators of states are general quantum operations instead of unitary operators. 1QFA with pure states and unitary evolutions usually have less recognition power than *deterministic finite automata* (DFA) due to the unitarity (reversibility) of quantum physics and the finite memory of finite automata. 1QFACL can accept all regular languages but their measurement is very complicated. However, one would expect a quantum variant to exceed (or at least to be not weaker than) the corresponding classical computing model, and such quantum computing models are practical and feasible as well. For this reason, we think that a quantum computer should inherit the characteristics of classical computers but further advance classical component by employing quantum mechanics principle.

Motivated by this idea, a new model of quantum automata including a classical component was outlined in [33]. Herein, we reformulate the definition of this new model of MO-1QFA, namely, 1QFA *together with classical states* (1QFAC, for short), and in particular, we inves-

tigate some of the basic properties of this new model. As MO-1QFA [23, 9], 1QFAC execute only a measurement for computing each input string, following the last symbol scanned. In this new model, we preserve the component of DFA that is used to control the choice of unitary transformation for scanning each input symbol. We now describe roughly a 1QFAC $\mathcal{A}$ computing an input string, delaying the details until Section 2.

At start up, automaton $\mathcal{A}$ is in an initial classical state and in an initial quantum state. By reading the first input symbol, the classical transformation results in a new classical state as current state, and, the initial classical state together with current input symbol assigns a unitary transformation to process the initial quantum state, leading to a new quantum state as current state. Afterwards, the machine reads the next input symbol, and similar to the above process, its classical state will be updated by reading the current input symbol and, at the same time, with the current classical state and input symbol, a new unitary transformation is assigned to execute the current quantum state. Subsequently, it continues to operate for the next step, until the last input symbol has been scanned. According to the last classical state, a measurement is assigned to perform on the final quantum state, producing a result of accepting or rejecting the input string.

Therefore, a 1QFAC performs only one measurement for computing each input string, doing so after reading the last symbol. However, the measurement is chosen according to the last classical state reached after scanning the input string. Thus, when a 1QFAC has only one classical state, it reduces to an MO-1QFA [23, 9]. On the one hand, 1QFAC model develops MO-1QFA by adding DFA's component, and on the other hand, 1QFAC advance DFA by employing the fundamentals of quantum mechanics.

We want to stress that 1QFAC are not the one-way version of *two-way finite automata with quantum and classical states* (2QCFA for short) proposed by Ambains and Watrous [3]. One of the differences is thatin the one-way version of 2QCFA, after the tape head reads an input symbol, either a measurement or a unitary transformation is performed, while in 1QFAC there is no intermediate measurement, and a single measurement is performed only after scanning the input string.

Though 1QFAC make only one measurement for computing each input string and the evolutions of states are unitary instead of general operations, the set of languages accepted by 1QFAC (with no error) consists precisely of all regular languages. Therefore, 1QFAC have more recognition power than MO-1QFA. As we know, the set of languages accepted by 1QFACL is constituted by all regular languages [24], but 1QFACL need measurement after reading each input symbol and the measurement is not only restricted to accepting, rejecting, and non-halting, but also other results related to the control language attached to the machine. Therefore, the computing process of a 1QFACL is usually much more complicated than that of a 1QFAC. On the other hand, measuring may lead to more errors for the machine.

It should be stressed that 1QFAC can accept some regular languages with exponentially less states than corresponding DFA [17], and there is no MO-1QFA [23], or MM-1QFA [19], or multi-letter 1QFA [6] that can accept these languages with bounded error. Hence, in a way, 1QFAC can be thought of as a more practical model of QFA, showing better state complexity than DFA due to its quantum computing component, and stronger recognition power of languages than MO-1QFA, MM-1QFA, and multi-letter 1QFA. Furthermore, for accepting the same regular language, we will show that 1QFAC have better state complexity than 1QFACL.

The main technical contributions of the paper contain five aspects. In Section 2, after reviewing some existing 1QFA models, we define 1QFAC formally and then prove that the set of languages accepted by 1QFAC with bounded error consists precisely of all regular languages. Here we used a technique of topological automata [18] to show this result. Indeed, we can also prove it with an alternative method in detail, based on a well-know technique for one-way probabilistic automata by Rabin [31] and already applied for MM-1QFA by Kondacs and Watrous [19], but the process is much longer, so we here use a technique of topological automata.

Then, in Section 3, we study the state complexity of 1QFAC. We show that, for any prime $m \geq 2$, there exists a regular language L_m whose minimal DFA needs $m+1$ states and neither MO-1QFA nor MM-1QFA can accept L_m, but there exists a 1QFAC accepting L_m with only two classical states and $O(\log(m))$ quantum basis states. In addition, we show that, for any $m \geq 2$, and any input string z, there exists a regular language $L_z(m)$ that can not be accepted by any multi-letter 1QFA or MO-1QFA, but there exists a 1QFAC $\mathcal{A}_m$ accepting it with only 2 classical states and $O(\log(m))$ quantum basis states. In contrast, the minimal DFA accepting $L_z(m)$ has $(|z|+1)m$ states, where $|z|$ denotes the length of z.

In Section 4, we study the equivalence problem of 1QFAC. Any two 1QFAC $\mathcal{A}_1$ and $\mathcal{A}_2$ over the same input alphabet Σ are equivalent (resp. k-equivalent) iff their probabilities for accepting any input string (resp. length not more than k) are equal. We reformulate any given 1QFAC with a bilinear computing machine. According to [31, 27, 36], it follows that 1QFAC $\mathcal{A}_1$ and $\mathcal{A}_2$ over the same input alphabet Σ are equivalent if and only if they are $(k_1n_1)^2+(k_2n_2)^2-1$-equivalent, and furthermore there exists a polynomial-time $O([(k_1n_1)^2+(k_2n_2)^2]^4)$ algorithm for determining their equivalence, where k_1 and k_2 are the numbers of classical states of $\mathcal{A}_1$ and $\mathcal{A}_2$, as well as n_1 and n_2 are the numbers of quantum basis states of $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively.

In general, notation used in this paper will be explained whenever new symbols appear. A language L over alphabet Σ is accepted by a computing model with *bounded error* if there exist $\lambda > 0$ and $\epsilon > 0$ such that the accepting probability for $x \in L$ is at least $\lambda + \epsilon$ and the accepting probability for $x \notin L$ is at most $\lambda - \epsilon$. In this paper, we always consider the accepting scheme of machines to be bounded error unless we emphasize otherwise.

2. One-way quantum finite automata together with classical states

In this section, we introduce the definition of 1QFAC and then prove its recognition power of languages. For the sake of readability, we first recall the definitions of MO-1QFA, MM-1QFA, multi-letter 1QFA, and 1QFACL.

2.1. Review of other one-way quantum finite automata

An MO-1QFA is defined as a quintuple $\mathcal{A} = (Q, \Sigma, |\psi_0\rangle, \{U(\sigma)\}_{\sigma\in\Sigma}, Q_{acc})$, where Q is a set of finite states, $|\psi_0\rangle$ is the initial state that is a superposition of the states in Q, Σ is a finite input alphabet, $U(\sigma)$ is a unitary matrix for each $\sigma \in \Sigma$, and $Q_{acc} \subseteq Q$ is the set of accepting states.

As usual, we identify Q with an orthonormal base of a complex Euclidean space and every state $q \in Q$ is identified with a basis vector, denoted by Dirac symbol $|q\rangle$ (a column vector), and $\langle q|$ is the conjugate transpose of $|q\rangle$. We describe the computing process for any given input string $x = \sigma_1\sigma_2\cdots\sigma_m \in \Sigma^*$. At the beginning the machine $\mathcal{A}$ is in the initial state $|\psi_0\rangle$, and upon reading σ_1, the transformation $U(\sigma_1)$ acts on $|\psi_0\rangle$. After that, $U(\sigma_1)|\psi_0\rangle$ becomes the current state and the machine reads σ_2. The process continues until the machine has read σ_m ending in the state $|\psi_x\rangle = U(\sigma_m)U(\sigma_{m-1})\cdots U(\sigma_1)|\psi_0\rangle$. Finally, a measurement is performed on $|\psi_x\rangle$ and the accepting probability $p_a(x)$ is equal to

$$p_a(x) = \langle\psi_x|P_a|\psi_x\rangle = \||P_a|\psi_x\rangle\|^2$$

where $P_a = \sum_{q\in Q_{acc}} |q\rangle\langle q|$ is the projection onto the subspace spanned by $\{|q\rangle : q \in Q_{acc}\}$.

Now we further recall the definition of multi-letter QFA [6].

A k-letter 1QFA $\mathcal{A}$ is defined as a quintuple $\mathcal{A} = (Q, \Sigma, |\psi_0\rangle, \nu, Q_{acc})$ where Q, $|\psi_0\rangle$, Σ, $Q_{acc} \subseteq Q$, are the same as those in MO-1QFA above, and ν is a function that assigns a unitary transition matrix U_w on $\mathbb{C}^{|Q|}$ for each string $w \in (\{\Lambda\} \cup \Sigma)^k$, where $|Q|$ is the cardinality of Q.

The computation of a k-letter 1QFA $\mathcal{A}$ works in the same way as the computation of an MO-1QFA, except that it applies unitary transformations corresponding not only to the last letter but the last k letters received. When $k = 1$, it is exactly an MO-1QFA as defined before. According to [6, 30], the languages accepted by k-letter 1QFA are a proper subset of regular languages for any k.

An MM-1QFA is defined as a 6-tuple $\mathcal{A} = (Q, \Sigma, |\psi_0\rangle, \{U(\sigma)\}_{\sigma\in\Sigma\cup\{\$\}}, Q_{acc}, Q_{rej})$, where $Q, Q_{acc} \subseteq Q, |\psi_0\rangle, \Sigma, \{U(\sigma)\}_{\sigma\in\Sigma\cup\{\$\}}$ are the same as those in an MO-1QFA defined above, $Q_{rej} \subseteq Q$ represents the set of rejecting states, and $\$ \notin \Sigma$ is a tape symbol denoting the right

end-mark. For any input string $x = \sigma_1\sigma_2\cdots\sigma_m \in \Sigma^*$, the computing process is similar to that of MO-1QFAs except that after every transition, $\mathcal{A}$ measures its state with respect to the three subspaces that are spanned by the three subsets Q_{acc}, Q_{rej}, and Q_{non}, respectively, where $Q_{non} = Q \setminus (Q_{acc} \cup Q_{rej})$. In other words, the projection measurement consists of $\{P_a, P_r, P_n\}$ where $P_a = \sum_{q\in Q_{acc}} |q\rangle\langle q|$, $P_r = \sum_{q\in Q_{rej}} |q\rangle\langle q|$, $P_n = \sum_{q\in Q\setminus(Q_{acc}\cup Q_{rej})} |q\rangle\langle q|$. The machine stops after the right end-mark \$ has been read. Of course, the machine may also stop before reading \$ if the current state, after the machine reading some σ_i $(1 \le i \le m)$, does not contain the states of Q_{non}. Since the measurement is performed after each transition with the states of Q_{non} being preserved, the accepting probability $p_a(x)$ and the rejecting probability $p_r(x)$ are given as follows (for convenience, we denote $\$ = \sigma_{m+1}$):

$$p_a(x) = \sum_{k=1}^{m+1} \|P_a U(\sigma_k) \prod_{i=1}^{k-1} (P_n U(\sigma_i))|\psi_0\rangle\|^2,$$

$$p_r(x) = \sum_{k=1}^{m+1} \|P_r U(\sigma_k) \prod_{i=1}^{k-1} (P_n U(\sigma_i))|\psi_0\rangle\|^2.$$

Here we define $\prod_{i=1}^{n} A_i = A_n A_{n-1} \cdots A_1$.

Bertoni *et al* [8] introduced a 1QFA, called 1QFACL that allows a more general measurement than the previous models. Similar to the case in MM-1QFA, the state of this model can be observed at each step, but an observable $\mathcal{O}$ is considered with a fixed, but arbitrary, set of possible results $\mathcal{C} = \{c_1, \ldots, c_n\}$, without limit to $\{a, r, g\}$ as in MM-1QFA. The accepting behavior in this model is also different from that of the previous models. On any given input word x, the computation displays a sequence $y \in \mathcal{C}^*$ of results of $\mathcal{O}$ with a certain probability $p(y|x)$, and the computation is accepted if and only if y belongs to a fixed regular language $\mathcal{L} \subseteq \mathcal{C}^*$. Bertoni *et al* [8] called such a language $\mathcal{L}$ *control language*.

More formally, given an input alphabet Σ and the end-marker symbol $\$ \notin \Sigma$, a 1QFACL over the working alphabet $\Gamma = \Sigma \cup \{\$\}$ is a five-tuple $\mathcal{M} = (Q, |\psi_0\rangle, \{U(\sigma)\}_{\sigma\in\Gamma}, \mathcal{O}, \mathcal{L})$, where

- Q, $|\psi_0\rangle$ and $U(\sigma)$ $(\sigma \in \Gamma)$ are defined as in the case of MM-1QFA;
- $\mathcal{O}$ is an observable with the set of possible results $\mathcal{C} = \{c_1, \ldots, c_s\}$ and the projector set $\{P(c_i) : i = 1, \ldots, s\}$ of which $P(c_i)$ denotes the projector onto the eigenspace corresponding to c_i;
- $\mathcal{L} \subseteq \mathcal{C}^*$ is a regular language (control language).

The input word w to 1QFACL $\mathcal{M}$ is in the form: $w \in \Sigma^*\$$, with symbol \$ denoting the end of a word. Now, we define the behavior of $\mathcal{M}$ on word $x_1 \ldots x_n\$$. The computation starts in the state $|\psi_0\rangle$, and then the transformations associated with the symbols in the word $x_1 \ldots x_n\$$ are applied in succession. The transformation associated with any symbol $\sigma \in \Gamma$ consists of two steps:

1. First, $U(\sigma)$ is applied to the current state $|\phi\rangle$ of $\mathcal{M}$, yielding the new state $|\phi^{'}\rangle = U(\sigma)|\phi\rangle$.

2. Second, the observable $\mathcal{O}$ is measured on $|\phi^{'}\rangle$. According to quantum mechanics principle, this measurement yields result c_k with probability $p_k = ||P(c_k)|\phi^{'}\rangle||^2$, and the state of $\mathcal{M}$ collapses to $P(c_k)|\phi^{'}\rangle/\sqrt{p_k}$.

Thus, the computation on word $x_1 \dots x_n\$$ leads to a sequence $y_1 \dots y_{n+1} \in \mathcal{C}^*$ with probability $p(y_1 \dots y_{n+1}|x_1 \dots x_n\$)$ given by

$$p(y_1 \dots y_{n+1}|x_1 \dots x_n\$) = \| \prod_{i=1}^{n+1} P(y_i)U(x_i)|\psi_0\rangle\|^2, \tag{1}$$

where we let $x_{n+1} = \$$ as stated before. A computation leading to the word $y \in \mathcal{C}^*$ is said to be accepted if $y \in \mathcal{L}$. Otherwise, it is rejected. Hence, the accepting probability of 1QFACL $\mathcal{M}$ is defined as:

$$\mathcal{P}_{\mathcal{M}}(x_1 \dots x_n) = \sum_{y_1 \dots y_{n+1} \in \mathcal{L}} p(y_1 \dots y_{n+1}|x_1 \dots x_n\$). \tag{2}$$

2.2. One-way quantum finite automata together with classical states

In Section 1, we gave the motivation for introducing 1QFAC. We now define formally the proposed model. To this end, we need the following notation. Given a finite set B, we denote by $\mathcal{H}(B)$ the Hilbert space freely generated by B. Furthermore, we denote by I and O the identity operator and zero operator on $\mathcal{H}(Q)$, respectively.

Definition 1. A 1QFAC $\mathcal{A}$ is defined by a 9-tuple

$$\mathcal{A} = (S, Q, \Sigma, \Gamma, s_0, |\psi_0\rangle, \delta, \mathbb{U}, \mathcal{M})$$

where:

- Σ is a finite set (the *input alphabet*);
- Γ is a finite set (the *output alphabet*);
- S is a finite set (the set of *classical states*);
- Q is a finite set (the *quantum state basis*);
- s_0 is an element of S (the *initial classical state*);
- $|\psi_0\rangle$ is a unit vector in the Hilbert space $\mathcal{H}(Q)$ (the *initial quantum state*);
- $\delta : S \times \Sigma \to S$ is a map (the *classical transition map*);

- $\mathbb{U} = \{U_{s\sigma}\}_{s\in S,\sigma\in\Sigma}$ where $U_{s\sigma} : \mathcal{H}(Q) \to \mathcal{H}(Q)$ is a unitary operator for each s and σ (the *quantum transition operator* at s and σ);

- $\mathcal{M} = \{\mathcal{M}_s\}_{s\in S}$ where each $\mathcal{M}_s$ is a projective measurement over $\mathcal{H}(Q)$ with outcomes in Γ (the *measurement operator at* s).

Hence, each $\mathcal{M}_s = \{P_{s,\gamma}\}_{\gamma\in\Gamma}$ such that $\sum_{\gamma\in\Gamma} P_{s,j} = I$ and $P_{s,\gamma}P_{s,\gamma'} = \begin{cases} P_{s,\gamma}, & \gamma = \gamma', \\ O, & \gamma \neq \gamma'. \end{cases}$ Furthermore, if the machine is in classical state s and quantum state $|\psi\rangle$ after reading the input string, then $\|P_{s,\gamma}|\psi\rangle\|^2$ is the probability of the machine producing outcome γ on that input.

Remark 2. Map δ can be extended to a map $\delta^* : \Sigma^* \to S$ as usual. That is, $\delta^*(s,\epsilon) = s$; for any string $x \in \Sigma^*$ and any $\sigma \in \Sigma$, $\delta^*(s,\sigma x) = \delta^*(\delta(s,\sigma),x)$.

Remark 3. A specially interesting case of the above definition is when $\Gamma = \{a, r\}$, where a denotes *accepting* and r denotes *rejecting*. Then, $\mathcal{M} = \{\{P_{s,a}, P_{s,r}\} : s \in S\}$ and, for each $s \in S$, $P_{s,a}$ and $P_{s,r}$ are two projectors such that $P_{s,a} + P_{s,r} = I$ and $P_{s,a}P_{s,r} = O$. In this case, $\mathcal{A}$ is an acceptor of languages over Σ.

For the sake of convenience, we denote the map $\mu : \Sigma^* \to S$, induced by δ, as $\mu(x) = \delta^*(s_0, x)$ for any string $x \in \Sigma^*$.

We further describe the computing process of $\mathcal{A} = (S, Q, \Sigma, s_0, |\psi_0\rangle, \delta, \mathbb{U}, Q_{\text{acc}})$ for input string $x = \sigma_1\sigma_2\cdots\sigma_m$ where $\sigma_i \in \Sigma$ for $i = 1, 2, \cdots, m$.

The machine $\mathcal{A}$ starts at the initial classical state s_0 and initial quantum state $|\psi_0\rangle$. On reading the first symbol σ_1 of the input string, the states of the machine change as follows: the classical state becomes $\mu(\sigma_1)$; the quantum state becomes $U_{s_0\sigma_1}|\psi_0\rangle$. Afterward, on reading σ_2, the machine changes its classical state to $\mu(\sigma_1\sigma_2)$ and its quantum state to the result of applying $U_{\mu(\sigma_1)\sigma_2}$ to $U_{s_0\sigma_1}|\psi_0\rangle$.

The process continues similarly by reading $\sigma_3, \sigma_4, \cdots, \sigma_m$ in succession. Therefore, after reading σ_m, the classical state becomes $\mu(x)$ and the quantum state is as follows:

$$U_{\mu(\sigma_1\cdots\sigma_{m-2}\sigma_{m-1})\sigma_m} U_{\mu(\sigma_1\cdots\sigma_{m-3}\sigma_{m-2})\sigma_{m-1}} \cdots U_{\mu(\sigma_1)\sigma_2} U_{s_0\sigma_1}|\psi_0\rangle. \tag{3}$$

Let $\mathcal{U}(Q)$ be the set of unitary operators on Hilbert space $\mathcal{H}(Q)$. For the sake of convenience, we denote the map $v : \Sigma^* \to \mathcal{U}(Q)$ as: $v(\epsilon) = I$ and

$$v(x) = U_{\mu(\sigma_1\cdots\sigma_{m-2}\sigma_{m-1})\sigma_m} U_{\mu(\sigma_1\cdots\sigma_{m-3}\sigma_{m-2})\sigma_{m-1}} \cdots U_{\mu(\sigma_1)\sigma_2} U_{s_0\sigma_1} \tag{4}$$

for $x = \sigma_1\sigma_2\cdots\sigma_m$ where $\sigma_i \in \Sigma$ for $i = 1, 2, \cdots, m$, and I denotes the identity operator on $\mathcal{H}(Q)$, indicated as before.

By means of the denotations μ and v, for any input string $x \in \Sigma^*$, after $\mathcal{A}$ reading x, the classical state is $\mu(x)$ and the quantum states $v(x)|\psi_0\rangle$.

Finally, the probability $\text{Prob}_{\mathcal{A},\gamma}(x)$ of machine $\mathcal{A}$ producing result γ on input x is as follows:

$$\text{Prob}_{\mathcal{A},\gamma}(x) = \|P_{\mu(x),\gamma} v(x)|\psi_0\rangle\|^2. \tag{5}$$

In particular, when $\mathcal{A}$ is thought of as an acceptor of languages over Σ ($\Gamma = \{a, r\}$), we obtain the probability $\text{Prob}_{\mathcal{A},a}(x)$ for accepting x:

$$\text{Prob}_{\mathcal{A},a}(x) = \|P_{\mu(x),a} v(x)|\psi_0\rangle\|^2. \tag{6}$$

For intuition, we depict the above process in Figure 1.

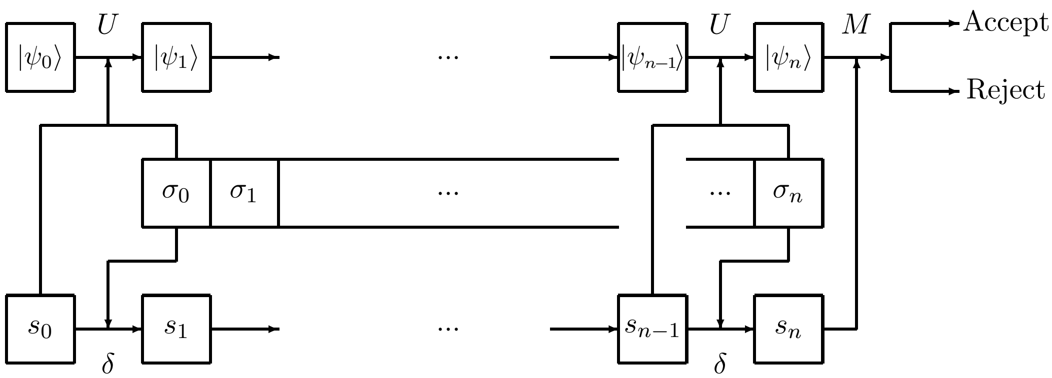

Figure 1: 1QFAC dynamics as an acceptor of language

Remark 4. If a 1QFAC $\mathcal{A}$ has only one classical state, then $\mathcal{A}$ reduces to an MO-1QFA [23]. Therefore, the set of languages accepted by 1QFAC with only one classical state is a proper subset of regular languages (exactly, the languages whose syntactic monoid is a group [9]). However, we prove that 1QFAC can accept all regular languages with no error.

Proposition 5. *Let Σ be a finite set. Then each regular language over Σ that is accepted by a minimal DFA of k states is also accepted by some 1QFAC with no error and with 1 quantum basis state and k classical states.*

Proof. Let $L \subseteq \Sigma^*$ be a regular language. Then there exists a DFA $M = (S, \Sigma, \delta, s_0, F)$ accepting L, where, as usual, S is a finite set of states, $s_0 \in S$ is an initial state, $F \subseteq Q$ is a set of accepting states, and $\delta : Q \times \Sigma \to Q$ is the transition function. We construct a 1QFAC $\mathcal{A} = (S, Q, \Sigma, \Gamma, s_0, |\psi_0\rangle, \delta, \mathbb{U}, \mathcal{M})$ accepting L without error, where S, Σ, s_0, and δ are the same as those in M, and, in addition, $\Gamma = \{a, r\}$, $Q = \{0\}$, $|\psi_0\rangle = |0\rangle$, $\mathbb{U} = \{U_{s\sigma} :$

$s \in S, \sigma \in \Sigma\}$ with $U_{s\sigma} = I$ for all $s \in S$ and $\sigma \in \Sigma$, $\mathcal{M} = \{\{P_{s,a}, P_{s,r}\} : s \in S\}$ assigned as: if $s \in F$, then $P_{s,a} = |0\rangle\langle 0|$ and $P_{s,r} = O$ where O denotes the zero operator as before; otherwise, $P_{s,a} = O$ and $P_{s,r} = |0\rangle\langle 0|$.

By the above definition of 1QFAC $\mathcal{A}$, it is easy to check that the language accepted by $\mathcal{A}$ with no error is exactly L. □

Remark 6. For any regular language L over $\{0, 1\}$ accepted by a k state DFA, it was proved that there exists a 1QFACL accepting L with no error and with $3k$ classical states ($3k$ is the number of states of its minimal DFA accepting the control language) and 3 quantum basis states [24]. Here, for 1QFAC, we require only k classical states and 1 quantum basis states. Therefore, in this case, 1QFAC have better state complexity than 1QFACL.

On the other hand, we show that any language accepted by a 1QFAC is regular. In order to show this we reduce a 1QFAC to a topological automaton and obtain the result as a corollary of a more general result about such automata [5, 18]. We start by recalling the relevant concepts concerning topological automata (for more details see [18]).

Definition 7. A *class of topological automata* is a tuple $(\mathcal{O}, \mathcal{C}, \mathcal{M})$ where:

- $\mathcal{O}$ is a topological monoid: $\mathcal{O}$ is a metric space and also a monoid for which the multiplication $\circ$ is a continuous map. $\mathcal{O}$ is called the set of operators.
- $\mathcal{C}$ is a metric space; $\mathcal{O}$ right acts continuously on $\mathcal{C}$. $\mathcal{C}$ is called the set of configurations. The action is denoted by $\cdot$.
- $\mathcal{M}$ is a set of continuous maps from $\mathcal{C}$ to $\mathbb{R}$. $\mathcal{M}$ is called the set of measures. Furthermore, for each $m \in \mathcal{M}$, and for each operator $o \in \mathcal{O}$, the map $m \circ o$ is in $\mathcal{M}$.

Several classic definitions of automata fit a class of topological automata, including DFA, probabilistic and MO-1QFA. The definition of topological automata is as expected.

Definition 8. A *deterministic topological automaton* of a class $(\mathcal{O}, \mathcal{C}, \mathcal{M})$ over an alphabet Σ is a triple $(c_0, (X_\sigma)_\sigma, m)$ where:

- c_0 is a configuration in $\mathcal{C}$ called the initial configuration;
- X_σ is an operator in $\mathcal{O}$ for each $\sigma \in \Sigma$;
- m is a measure in $\mathcal{M}$.

Remark 9. A 1QFAC acceptor $\mathcal{A} = (S, Q, \Sigma, \Gamma, s_0, |\psi_0\rangle)$ over Σ with $|S| = k$ and $|Q| = n$ can be seen as a topological automaton $(c_0, (X_\sigma)_\sigma, m)$ over the alphabet Σ and class $(\mathcal{O}, \mathcal{C}, \mathcal{M})$ where:

- $\mathcal{C}$ is a subset of the real vector space of dimension kn whose elements are k-vectors $c = (|\varphi_i\rangle)_{i=0\ldots k-1}$ where each $|\varphi_i\rangle$ is a vector of dimension n and $|\varphi_i\rangle$=0 for all $i = 0\ldots k-1$ except for one such i, and for that $|\varphi_i\rangle$ is a unit vector. Clearly, $\mathcal{C}$ is a metric space since it inherits the metric structure of the vector space.

- $\mathcal{O}$ is a subset of the linear transformations in $\mathcal{C}$ represented by a matrix $kn \times kn$ where each column (seen as a matrix $k \times k$,) contains only one non-zero $n \times n$ matrix, this matrix being unitary. Clearly, this space is a topological monoid, since: (i) it is a metric space (inhering the metric of the linear transformations); (ii) it is closed for composition; and (iii) the identity belongs to $\mathcal{O}$. Moreover, $\mathcal{O}$ acts continuously in $\mathcal{C}$, since each $f \in \mathcal{O}$ is linear, and furthermore, $f(c) \in \mathcal{C}$ for all $c \in \mathcal{C}$ and $f \in \mathcal{O}$.

- $\mathcal{M}$ is the set of continuous maps obtained by computing the square of the Euclidean norm after applying a projector in $\mathcal{C}$.

- $c_0 = (|\varphi_{0i}\rangle)_{i=0\ldots k-1}$ where $|\varphi_{0i}\rangle = 0$ for $i = 1\ldots k-1$ and $|\varphi_{00}\rangle = |\psi_0\rangle$.

- X_σ is the $kn \times kn$ matrix such that at column $i+1$ and row $j+1$ (seen as a $k \times k$ matrix), with $s_j = \delta(s_i, \sigma)$, we place the $n \times n$ unitary matrix $U_{s_i\sigma}$, and the remaining part of column $i+1$ is 0, for $i = 0\ldots k-1$.

- $m(c) = \langle c|P|c\rangle$ where P is the projector represented by a $kn \times kn$ matrix such that at column $i+1$ and row $i+1$ (seen as a $k \times k$ matrix) has the $n \times n$ matrix $P_{s_i,a}$ and is 0 elsewhere.

It is easy to verify that $\|P_{\mu(x),a}v(x)|\psi_0\rangle\|^2 = \mathrm{Prob}_{\mathcal{A},a}(x) = m(X_x.c_0)$. Observe that the closure of the set $\{X_x, x \in \Sigma^*\}$ is compact since the operators X_x are bounded (indeed the ℓ_2 norm of all X_i is bounded by $\sqrt{n}$). The following result implies that the language recognized by a 1QFAC acceptor is regular.

Theorem 10 ([18])**.** *Let L be a language recognized with bounded error by a topological automaton $(c_0, (X_\sigma)_\sigma, m)$. If $\overline{\{X_x, x \in \Sigma^*\}}$ is compact (that is, the monoid generated by the X_i is relatively compact), then L is regular.*

Remark 11. Indeed, without using the above theorem regarding topological automata, we can also prove that the languages recognized by 1QFAC are regular with an alternative method in detail, based on a well-know idea for one-way probabilistic automata by Rabin [31], that was already applied for MM-1QFA by Kondacs and Watrous [19] as well as for MO-1QFA by Brodsky and Pippenger [9]. However, the process is much longer and some different techniques are needed, since both classical and quantum states are involved in 1QFAC. For the sake of simplicity, we only present here the technique based on topological automata.

3. State complexity of 1QFAC

State complexity of classical finite automata has been an intriguing field with important practical applications [32, 37]. In this section, we show certain advantages of 1QFAC over DFA and other 1QFA models. Although 1QFAC accept only regular languages as DFA, 1QFAC can accept some languages with essentially less number of states than DFA and these languages can *not* be accepted by any MO-1QFA and MM-1QFA as well as multi-letter 1QFA. In this section, our purpose is to prove these claims.

First, we establish a technical result concerning the acceptability by 1QFAC of languages resulting from set operations on languages accepted by MO-1QFA and by DFA.

Lemma 12. *Let Σ be a finite alphabet. Suppose that the language L_1 over Σ is accepted by a minimal DFA with n_1 states and the language L_2 over Σ is accepted by an MO-1QFA with n_2 quantum basis states with bounded error ϵ. Then the intersection $L_1 \cap L_2$, union $L_1 \cup L_2$, differences $L_1 \setminus L_2$ and $L_2 \setminus L_1$ can be accepted by some 1QFAC with n_1 classical states and n_2 quantum basis states with bounded error ϵ.*

Proof. Let $A_1 = (S, \Sigma, \delta, s_0, F)$ be a minimal DFA accepting L_1, and let $A_2 = (Q, \Sigma, |\psi_0\rangle, \{U(\sigma)\}_{\sigma\in\Sigma}, Q_{acc})$ be an MO-1QFA accepting L_2, where $s_0 \in S$ is the initial state, δ is the transition function, and $F \subseteq S$ is a finite subset denoting accepting states; the symbols in A_2 are the same as those in the definition of MO-1QFA as above.

Then by A_1 and A_2 we define a 1QFAC $\mathcal{A} = (S, Q, \Sigma, \Gamma, s_0, |\psi_0\rangle, \delta, \mathbb{U}, \mathcal{M})$ accepting $L_1 \cap L_2$, where $S, Q, \Sigma, s_0, |\psi_0\rangle, \delta$ are the same as those in A_1 and A_2, $\Gamma = \{a, r\}$, $\mathbb{U} = \{U_{s\sigma} = U(\sigma) : s \in S, \sigma \in \Sigma\}$, and $\mathcal{M} = \{M_s : s \in S\}$ where $M_s = \{P_{s,a}, P_{s,r}\}$ and

$$P_{s,a} = \begin{cases} \sum_{p\in Q_{acc}} |p\rangle\langle p|, & s \in F; \\ O, & s \notin F, \end{cases}$$

where O denotes the zero operator, and $P_{s,r} = I - P_{s,a}$ with I being the identity operator.

According to the above definition of 1QFAC, we easily know that, for any string $x \in \Sigma^*$, if $x \in L_1$ then the accepting probability of 1QFAC $\mathcal{A}$ is equal to the accepting probability of MO-1QFA A_2; if $x \notin L_1$ then the accepting probability of 1QFAC $\mathcal{A}$ is zero. So, 1QFAC $\mathcal{A}$ accepts the intersection $L_1 \cap L_2$.

Similarly, we can construct the other three 1QFAC accepting the union $L_1 \cup L_2$, differences $L_1 \setminus L_2$, and $L_2 \setminus L_1$, respectively. Indeed, we only need define different measurements in these 1QFAC. If we construct 1QFAC accepting $L_1 \cup L_2$, then

$$P_{s,a} = \begin{cases} I, & s \in F; \\ \sum_{p\in Q_{acc}} |p\rangle\langle p|, & s \notin F. \end{cases}$$

If we construct 1QFAC accepting $L_1 \setminus L_2$, then

$$P_{s,a} = \begin{cases} \sum_{p \in Q \setminus Q_{acc}} |p\rangle\langle p|, & s \in F; \\ O, & s \notin F. \end{cases}$$

If we construct 1QFAC accepting $L_2 \setminus L_1$, then

$$P_{s,a} = \begin{cases} \sum_{p \in Q \setminus Q_{acc}} |p\rangle\langle p|, & s \notin F; \\ O, & s \in F. \end{cases}$$

□

Now we consider a regular language $L^0(m) = \{w0 : w \in \Sigma^*, |w0| = km, k = 1, 2, 3, \cdots\}$ where $\Sigma = \{0, 1\}$. Clearly, the minimal classical DFA accepting $L^0(m)$ has $m + 1$ states, as depicted in Figure 2.

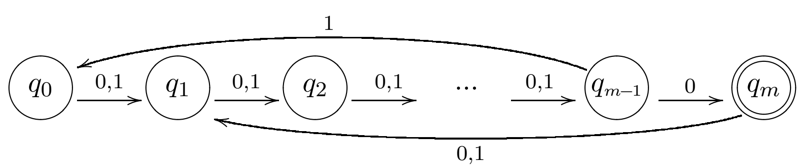

Figure 2: DFA accepting L_m^0

Indeed, neither MO-1QFA nor MM-1QFA can accept $L^0(m)$. We can easily verify this result by employing a lemma from [9, 15]. That is,

Lemma 13 ([9, 15])**.** *Let L be a regular language, and let M be its minimal DFA containing the construction in Figure 3, where states p and q are distinguishable (i.e., there exists a string z such that either $\delta(p, z)$ or $\delta(q, z)$ is an accepting state). Then, L can not be accepted by MM-1QFA.*

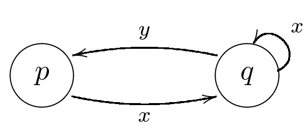

Figure 3: Construction not accepted by an MM-1QFA

Proposition 14. *Let $\Sigma = \{0, 1\}$. Then neither MO-1QFA nor MM-1QFA can accept $L^0(m)$.*

Proof. Indeed, it suffices to show that no MM-1QFA can accept $L^0(m)$ since the languages accepted by MO-1QFA are also accepted by MM-1QFA [2, 9, 8]. By Lemma 13, we know that $L^0(m)$ can not be accepted by any MM-1QFA since its minimal DFA (see Figure 2) contains such a construction: For example, we can take $p = q_0, q = q_m, x = 0^m, y = 0^{m-1}1, z = \epsilon$. □

Proposition 15. *$L^0(m)$ can be accepted by a 1QFAC with 2 classical states and 2 quantum basis states.*

Proof. Let the languages $L^0 = \{w0 : w \in \Sigma^*\}$ and $L(m) = \{w : w \in \Sigma^*, |w| = km, k = 1, 2, 3, \cdots\}$. Then L^0 can be accepted by a minimal DFA A_1 with 2 states, and $L(m)$ can be accepted by an MO-1QFA A_2 with 2 quantum basis states. Indeed, A_1 can be clearly described by Figure 4. A_2 can be defined as $A_2 = (Q, \Sigma, |\psi_0\rangle, \mathbb{U}, Q_{acc})$ where $Q = \{0, 1\}$, $|\psi_0\rangle = |0\rangle$, $\mathbb{U} = \{U_0, U_1\}$ with

$$U_0 = U_1 = \begin{bmatrix} \cos\frac{\pi}{m} & -\sin\frac{\pi}{m} \\ \sin\frac{\pi}{m} & \cos\frac{\pi}{m} \end{bmatrix},$$

and $Q_{acc} = \{0\}$. Therefore, the measurement consists of two projection operators $P_a = |0\rangle\langle 0|$ and $P_r = |1\rangle\langle 1|$. It is clear that if $x \in L_2(m)$, then x can be accepted with no error, and if $x \notin L_2(m)$ the probability for accepting x is at most $\cos^2\frac{\pi}{m}$.

According to Lemma 12, there exists a 1QFAC $\mathcal{A}(m)$ accepting

$$L^0 \cap L(m) = L^0(m) = \{w0 : w \in \Sigma^*, |w0| = km, k = 1, 2, 3, \cdots,\}$$

such that if $x \in L^0(m)$, then x can be accepted with no error, and if $x \notin L^0(m)$ the probability for accepting x is at most $\cos^2\frac{\pi}{m}$, and the number of classical states in $\mathcal{A}(m)$ is 2 and the number of quantum basis states is also 2. $\square$

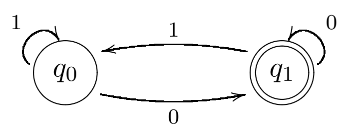

Figure 4: DFA accepting $\{0, 1\}^*0$

In the above proposition, we note that if m goes to infinity, then the probability for rejecting $x \notin L^0(m)$ goes to 0, which is not what we desire. Indeed, by using the technique in [2], we can overcome this problem. Let us recall an result from [2].

Proposition 16 ([2])**.** *Let the language $L_p = \{a^i : i$ is divisible by $p\}$ where p is a prime number. Then for any $\varepsilon > 0$, there exists an MM-1QFA with $O(\log(p))$ states such that for any $x \in L_p$, x is accepted with no error, and the probability for accepting $x \notin L_p$ is smaller than ε.*

Indeed, from the proof of Proposition 16 by [2], also as Ambainis and Freivalds pointed out in [2] (before Section 2.2 in [2]), Proposition 16 holds for MO-1QFA as well.

Clearly, by the same technique as the proof of Proposition 16 [2], then one can obtain that, by replacing L_p with $L(m) = \{w : w \in \Sigma^*, |w| = km, k = 1, 2, 3, \cdots\}$ with m being

a prime number, Proposition 16 still holds (by viewing all input symbols in Σ as a). By combining Proposition 16 with Lemma 12, we have the following corollary.

Corollary 17. *Suppose that m is a prime number. Then for any $\varepsilon > 0$, there exists a 1QFAC with 2 classical states and $O(\log(m))$ quantum basis states such that for any $x \in L^0(m)$, x is accepted with no error, and the probability for accepting $x \notin L^0(m)$ is smaller than ε.*

In summary, we have the following result.

Proposition 18. *For any prime number $m \geq 2$, there exists a regular language $L^0(m)$ satisfying: (1) neither MO-1QFA nor MM-1QFA can accept $L^0(m)$; (2) the number of states in the minimal DFA accepting $L^0(m)$ is $m + 1$; (3) for any $\varepsilon > 0$, there exists a 1QFAC with 2 classical states and $O(\log(m))$ quantum basis states such that for any $x \in L^0(m)$, x is accepted with no error, and the probability for accepting $x \notin L^0(m)$ is smaller than ε.*

One should ask at this point whether similar results can be established for multi-letter 1QFA as proposed by Belovs et al. [6].

Recall that 1-letter 1QFA is exactly an MO-1QFA. Any given k-letter QFA can be simulated by some $k+1$-letter QFA. However, Qiu and Yu [30] proved that the contrary does not hold. Belovs et al. [6] have already showed that $(a+b)^*b$ can be accepted by a 2-letter QFA but, as proved in [19], it cannot be accepted by any MM-1QFA. On the other hand, a^*b^* can be accepted by MM-1QFA [2] but it can not be accepted by any multi-letter 1QFA [30], and furthermore, there exists a regular language that can not be accepted by any MM-1QFA or multi-letter 1QFA [30].

To conclude this section, we will prove that, for any $m \geq 2$, there exists a regular language that can not accepted by any multi-letter 1QFA or any MO-1QFA, but there exists 1QFAC $\mathcal{A}_m$ accepting the language with 2 classical states and 2 quantum basis states. In addition, the minimal DFA accepting this language has $O(m)$ states.

Let Σ be an alphabet. For string $z = z_1 \cdots z_n \in \Sigma^*$, consider the regular language

$$L_z = \Sigma^* z_1 \Sigma^* z_2 \Sigma^* \cdots \Sigma^* z_n \Sigma^*.$$

(L_z belongs to piecewise testable set that was introduced by Simon [35] and studied in [28]. Brodsky and Pippenger [9] proved that L_z can be accepted by an MM-1QFA with $2n+3$ states.) Let another regular language $L(m) = \{w : w \in \Sigma^*, |w| = km, k = 1, 2, \cdots\}$. Then the minimal DFA accepting L_z needs $n+1$ states, and there exists an MO-1QFA accepting $L(m)$ with 2 quantum basis states. As a result, according to Lemma 12, the intersection $L_z(m)$ of L_z and $L(m)$ can be accepted by a 1QFAC with $n+1$ classical states and 2 quantum basis states. However, the minimal DFA accepting $L_z(m)$ needs $m(n+1)$ states and, in addition, we will prove that no multi-letter 1QFA can accept $L_z(m)$. The minimal DFA accepting $L_z(m)$

can be described by $A = (Q, \Sigma, \delta, q_0, F)$ where $Q = \{S_{ij} : i = 0, 1, \ldots, n; j = 1, 2, \ldots, m\}$, $\Sigma = \{z_1, z_2, \ldots, z_n\}$, $q_0 = S_{01}$, $F = \{S_{n1}\}$, and the transition function δ is defined as:

$$\delta(S_{ij}, \sigma) = \begin{cases} S_{n,(j \bmod m)+1}, & \text{if } i = n, \\ S_{i+1,(j \bmod m)+1}, & \text{if } i \neq n \text{ and } \sigma = z_{i+1}, \\ S_{i,(j \bmod m)+1}, & \text{if } i \neq n \text{ and } \sigma \neq z_{i+1}. \end{cases} \tag{7}$$

Also, the minimal DFA A can be depicted as Figure 5.

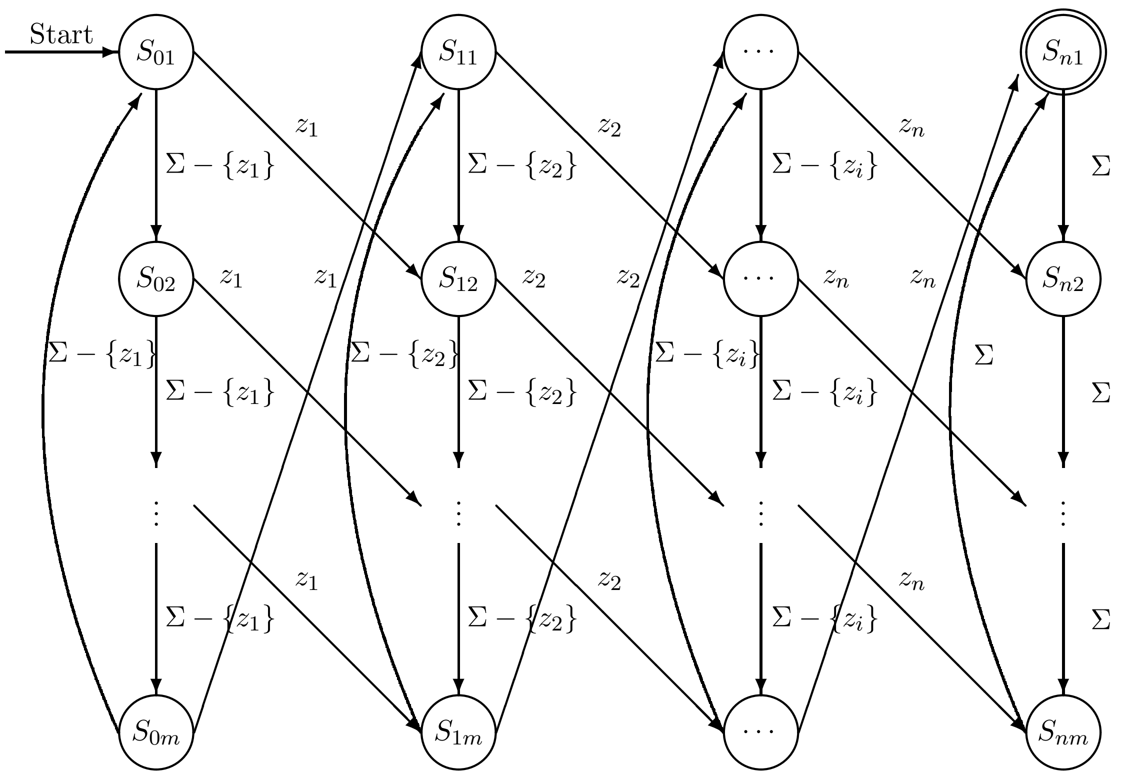

Figure 5: The transition figure of the minimal DFA accepting $L_z(m)$.

The number of states of the minimal DFA accepting $L_z(m)$ is $m(n + 1)$.

For the sake of simplicity, we consider a special case: $m = 2$, $n = 1$, and $\Sigma = \{0, 1\}$. Indeed, this case can also show the above problem as desired. So, we consider the following language:

$$L_0(2) = \{w : w \in \{0, 1\}^*0\{0, 1\}^*, |w| = 2k, k = 1, 2, \cdots\}.$$

The minimal DFA accepting $L_0(z)$ above needs 4 states and its transition figure is depicted by Figure 6 as follows.

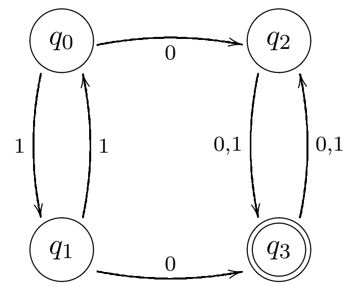

Figure 6: DFA accepting $w \in \{0,1\}^*0\{0,1\}^*$ with $|w|$ even.

We recall the definition of F-construction and a proposition from [6].

Definition 19 ([6]). A DFA with state transition function δ is said to *contain an F-construction* (see Figure 7) if there are non-empty words $t, z \in \Sigma^+$ and two distinct states $q_1, q_2 \in Q$ such that $\delta^*(q_1, z) = \delta^*(q_2, z) = q_2$, $\delta^*(q_1, t) = q_1$, $\delta^*(q_2, t) = q_2$, where $\Sigma^+ = \Sigma^* \backslash \{\epsilon\}$, ϵ denotes empty string.

We can depict F-construction by Figure 7.

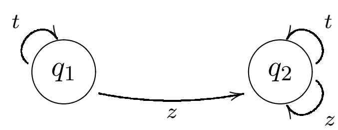

Figure 7: F-Construction

Lemma 20 ([6]). *A language L can be accepted by a multi-letter 1QFA with bounded error if and only if the minimal DFA of L does not contain any F-construction.*

In Figure 6, there are an F-construction: For example, we consider q_0 and q_3, and strings 00 and 11, from the above proposition which shows that no multi-letter 1QFA can accept $L_0(2)$.

According to Lemma 12, there exists a 1QFAC accepting the $L_0(2)$ with 2 classical states and 2 quantum basis states. However, from Figure 6 we know that the minimal DFA accepting $L_0(2)$ has 4 states.

In general, we have the following conclusion.

Proposition 21. *For any $m \geq 2$, there exists a regular language $L_0(m)$ that can not be accepted by any multi-letter 1QFA, but there exists a 1QFAC $\mathcal{A}_m$ accepting $L_0(m)$ with 2 classical states and 2 quantum basis states. In contrast, the minimal DFA accepting $L_0(m)$ has $O(m)$ states.*

If language $L_z(m)$ is considered, then one can obtain a more general result.

Proposition 22. *For any $m \geq 2$ and any string z with $|z| = n \geq 1$, then there exists a regular language $L_z(m)$ that can not be accepted by any multi-letter 1QFA, but there exists a 1QFAC $\mathcal{A}_m$ accepting $L_z(m)$ with $n+1$ classical states (independent of m) and 2 quantum basis states. In contrast, the minimal DFA accepting $L_z(m)$ has $m(n+1)$ states.*

Remark 23. Similarly to the case of Proposition 18, it is possible to extend the result of Proposition 22 to fixed bounded error, using precisely the same technique. In this case we have to restrict m to be a prime number. Then for any string z with $|z| = n \geq 1$ there exists a regular language $L_z(m)$ that can not be accepted by any multi-letter 1QFA, but for every ε there exists a 1QFAC $\mathcal{A}_m$ with $n+1$ classical states (independent of m) and $O(\log(m))$ quantum basis states such that if $x \in L_z(m)$, x is accepted with no error, and the probability for accepting $x \notin L_z(m)$ is smaller than ε. In contrast, the minimal DFA accepting $L_z(m)$ has $m(n+1)$ states.

4. Determining the equivalence of 1QFAC

In this section, we consider the equivalence problem of 1QFAC. For any given 1QFAC $\mathcal{A}_1$ and 1QFAC $\mathcal{A}_2$ over the same finite input alphabet Σ and finite output alphabet Γ, our purpose is to determine whether or not they are equivalent according to the following definition.

Definition 24. A 1QFAC $\mathcal{A}_1$ and another 1QFAC $\mathcal{A}_2$ over the same input alphabet Σ and output alphabet Γ are said to be *equivalent* (resp. *t-equivalent*) if $\mathrm{Prob}_{\mathcal{A}_1,\gamma}(w) = \mathrm{Prob}_{\mathcal{A}_2,\gamma}(w)$ for any $w \in \Sigma^*$ (resp. for any input string w with $|w| \leq t$) and any $\gamma \in \Gamma$.

In the following, we will present a methods to determine whether or not any two 1QFAC are equivalent by taking into account the reformulation of 1QFAC as topological automata (see Remark 9). For readability, we recall a mathematical model which is not an actual computing model but generalizes many classical computing models, including probabilistic automata [31, 27] and deterministic finite automata [17].

Definition 25. A *bilinear machine* (BLM) over the alphabet Σ is a tuple

$$\mathcal{M} = (S, \pi, \{M(\sigma)\}_{\sigma\in\Sigma}, \eta),$$

where S is a finite state set with $|S| = n$, $\pi \in \mathbb{C}^{1\times n}$, $\eta \in \mathbb{C}^{n\times 1}$ and $M(\sigma) \in \mathbb{C}^{n\times n}$ for $\sigma \in \Sigma$.

Associated to a BLM $\mathcal{M}$, the *word function* $f_{\mathcal{M}} : \Sigma^* \to \mathbb{C}$ is defined in the way: $f_{\mathcal{M}}(w) = \pi M(w_1)\dots M(w_m)\eta$, where $w = w_1 \dots w_m \in \Sigma^*$.

Definition 26. Two BLM $\mathcal{M}_1$ and $\mathcal{M}_2$ over the same alphabet Σ are said to be equivalent (resp. k-equivalent) if $f_{\mathcal{M}_1}(w) = f_{\mathcal{M}_2}(w)$ for any $w \in \Sigma^*$ (resp. for any input string w with $|w| \leq k$).

As indicated in [21], if we refer to [27, 36], then we can find that one can get a general result as follows.

Proposition 27 ([27, 36])**.** *Two BLM $\mathcal{A}_1$ and $\mathcal{A}_2$ with n_1 and n_2 states, respectively, are equivalent if and only if they are (n_1+n_2-1)-equivalent. Furthermore, there exists a polynomial-time algorithm running in time $O((n_1+n_2)^4)$ that takes as input two BLM $\mathcal{A}_1$ and $\mathcal{A}_2$ and determines whether $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent.*

Therefore, if we can simulate any 1QFAC by an equivalent BLM, then the equivalence problem of 1QFAC can be solved. Indeed, we can do that with two steps. First, we reformulate a 1QFAC as a topological automaton (as presented in Remark 9), then by bilinearization we simulate the resulting topological automaton by an equivalent BLM. More specifically, let $\mathcal{A}=(S,Q,\Sigma,\Gamma,s_0,|\psi_0\rangle,\delta,\mathbb{U},\mathcal{M})$ be a 1QFAC, where $S=\{s_0,s_1,\ldots,s_{k-1}\}$, and $n=|Q|$. According to the above technical method, we can design an equivalent topological automaton, say, $TP(\mathcal{A})=(c_0,(X_\sigma)_\sigma,m)$, such that, for any string $x\in\Sigma^*$,

$$\|P_{\mu(x),\gamma}v(x)|\psi_0\rangle\|^2=\mathrm{Prob}_{\mathcal{A},\gamma}(x)=m(X_x.c_0)=c_0^\dagger.X_x^\dagger.P.X_x.c_0. \tag{8}$$

Then by bilinearization technique, we have

$$\begin{aligned}
c_0^\dagger.X_x^\dagger.P.X_x.c_0 &= \|c_0.X_xP\|^2 \\
&= \sum_{i,j}|c_0.X_x|\psi_{s_i,j}\rangle|^2 \qquad (9)\\
&= (c_0\otimes c_0^*).(X_x\otimes X_x^*)\sum_{i,j}|\psi_{s_i,j}\rangle\otimes(|\psi_{s_i,j}\rangle)^* \qquad (10)
\end{aligned}$$

where, as we know

$$P=\begin{bmatrix} P_{s_0,\gamma} & 0 & \cdots & 0 \\ 0 & P_{s_1,\gamma} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_{s_{k-1},\gamma}\end{bmatrix},$$

$\{|\psi_{s_i,j}\rangle : j=1,2,\cdots,n\}$ is an orthonormal base corresponding to those columns of $P_{s_i,\gamma}$ in P.

Therefore, 1QFAC $\mathcal{A}$ can be equivalently simulated by a BLM, denoted by

$$BLM(\mathcal{A})=(S_{\mathcal{A}},c_0\otimes c_0^*,\{X_\sigma\otimes X_\sigma^*\}_{\sigma\in\Sigma},\sum_{i,j}|\psi_{s_i,j}\rangle\otimes(|\psi_{s_i,j}\rangle)^*) \tag{11}$$

where $|S_{\mathcal{A}}|=kn$ with k and n being the numbers of classical and quantum basis states, respectively.

By Proposition 27, we can obtain that two 1QFAC $\mathcal{A}_1 = (S_1, Q_1, \Sigma, \Gamma, s_0, |\psi_0^{(1)}\rangle, \delta_1, \mathbb{U}_1, \mathcal{M}_1)$ and $\mathcal{A}_2 = (S_2, Q_2, \Sigma, \Gamma, t_0, |\psi_0^{(2)}\rangle, \delta_2, \mathbb{U}_2, \mathcal{M}_2)$ are equivalent if and only if they are $(k_1n_1)^2 + (k_2n_2)^2 - 1$-equivalent, where k_i and n_i are the numbers of classical and quantum basis states of $\mathcal{A}_i$, respectively, $i = 1, 2$. In addition, there exists a polynomial-time algorithm running in time $O([(k_1n_1)^2 + (k_2n_2)^2]^4)$ that takes as input two 1QFAC $\mathcal{A}_1$ and $\mathcal{A}_2$ and determines whether $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent. We formulate this result as follows.

Theorem 28. *Two 1QFAC $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent if and only if they are $(k_1n_1)^2 + (k_2n_2)^2 - 1$-equivalent. Furthermore, there exists a polynomial-time algorithm running in time $O([(k_1n_1)^2 + (k_2n_2)^2]^4)$ that takes as input two 1QFAC $\mathcal{A}_1$ and $\mathcal{A}_2$ and determines whether $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent, where k_i and n_i are the numbers of classical and quantum basis states of $\mathcal{A}_i$, respectively, $i = 1, 2$.*

Remark 29. The above equivalence result and the existence of polynomial time algorithm are derived from Proposition 27 by using a bilinearization method. However, we can also directly prove the equivalence result and design a polynomial time algorithm as [20, 21] for determining the equivalence of quantum sequential machines and MM-1QFA. However, since the process is much more complicated and different techniques are needed, we choose to present only the bilinearization method.

5. Conclusions and problems

In this paper, we have developed in detail a new one-way QFA model, namely 1QFAC. 1QFAC can accept all regular languages with no error, and, in particular, 1QFAC can accept some languages with essentially less number of states than one-way DFA, but no MO-1QFA or MM-1QFA or multi-letter 1QFA can accept these languages. 1QFAC contain both classical and quantum components. By virtue of the classical component, 1QFAC can accept all regular languages. Due to the quantum component, 1QFAC can accept a subclass of regular languages with constant numbers of classical states and quantum basis states but, in contrast, the minimal DFA require non constant numbers of states. Therefore, 1QFAC have inherited the characteristics of DFA but improved on them by employing quantum computing. From the viewpoint of computing process, 1QFAC may also clearly be realized physically, and therefore this is a more practical model of quantum computing with finite memory.

The technical contributions of the paper are threefold: (1) We have proved that the set of languages accepted by 1QFAC consists precisely of all regular languages. (2) We have established that, for any $m \geq 2$, there exist some regular languages L_m whose minimal DFA needs $O(m)$ states, and no MO-1QFA or MM-1QFA or multi-letter 1QFA can accept L_m, but there exists 1QFAC accepting L_m with only constant classical states (independent of m) and two quantum basis states. In particular, when m is restricted to be a prime number, we can fix

the bounded error by employing the result of Ambainis and Freivalds [2]. (3) We have proved that any two 1QFAC $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent if and only if they are $(k_1n_1)^2+(k_2n_2)^2-1$-equivalent, where k_1 and k_2 are the numbers of classical states of $\mathcal{A}_1$ and $\mathcal{A}_2$, as well as n_1 and n_2 are the numbers of quantum basis states of $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively; in addition, there exists a polynomial-time $O(((k_1n_1)^2+(k_2n_2)^2)^4)$ algorithm for determining their equivalence.

To conclude, we would like to propose some problems for further consideration.

- State complexity of 1QFAC: For any given regular language L, if the minimal number of states of the DFA accepting L is n, then for any $n_1 < n$, whether or not there exists a 1QFAC accepting L with n_1 classical states and some quantum basis states?
- What would be the consequences of relaxing the notion of equivalence between automata to equivalence up to ε? More precisely, for instance, one should investigate the equivalence problem when two automata are considered equivalent iff their acceptance probability distributions over the strings do not differ more than ε at each string.
- 1QFA *with control languages* (1QFACL), the ancilla 1QFA in [26], and the Ciamarra 1QFA in [10] also accept all regular languages [22], and Remark 6 shows a certain advantage of 1QFAC over 1QFACL in state complexity. Compare the state complexity of 1QFAC with these 1QFA in detail, and how to simulate 1QFAC by these 1QFA and vice-versa?
- Give regular languages that are accepted by MO-1QFA and 1QFAC, but 1QFAC have both essentially less quantum basis states than MO-1QFA and essentially less classical states than their minimal DFA.
- If 1QFAC are allowed to be measured many times, i.e, measurement performed after reading each symbol like MM-1QFA, instead only the last symbol, then how about the recognition power of 1QFAC?

References

[1] A. Ambainis, M. Beaudry, M. Golovkins, A. Kikusts, M. Mercer, D. Thérien, Algebraic results on quantum automata, Theory of Computing Systems 39 (2006) 165-188.

[2] A. Ambainis, R. Freivalds, One-way quantum finite automata: strengths, weaknesses and generalizations, in: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Palo Alfo, California, USA, 1998, pp. 332-341.

[3] A. Ambainis, J. Watrous, Two-way finite automata with quantum and classical states, Theoretical Computer Science 287 (2002) 299-311.

[4] C. H. Bennett, Logical reversibility of computation, IBM J. Res. Dev. 17 (1973) 525-532.

[5] S. Bozapalidis, Extending stochasic and quantum functions, Theory Computing Systems 36 (2003) 183-197.

[6] A. Belovs, A. Rosmanis, and J. Smotrovs, Multi-letter Reversible and Quantum Finite Automata, in: Proceedings of the 13th International Conference on Developments in Language Theory (DLT'2007), Lecture Notes in Computer Science, Vol. 4588, Springer, Berlin, 2007, pp. 60-71.

[7] P. Benioff, The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines, Journal of Statistic Physics 22 (1980) 563-591.

[8] A. Bertoni, C. Mereghetti, B. Palano, Quantum Computing: 1-Way Quantum Automata, in: Proceedings of the 9th International Conference on Developments in Language Theory (DLT'2003), Lecture Notes in Computer Science, Vol. 2710, Springer, Berlin, 2003, pp. 1-20.

[9] A. Brodsky, N. Pippenger, Characterizations of 1-way quantum finite automata, SIAM Journal on Computing 31 (2002) 1456-1478.

[10] M. P. Ciamarra, Quantum Reversibility and a New Model of Quantum Automaton, in Proceeding of 13th International Symposium on Fundamentals of Computation Theory, Lecture Notes in Computer Science, Vol. 2138, Springer-Verlag, Berlin, 2001, pp. 376-379.

[11] D. Deutsh, Quantum theory, the Church-Turing principle and the universal quantum computer, Proceedings of the Royal Society of London Series A 400 (1985) 97-117.

[12] R.P. Feynman, Simulating physics with computers, International Journal of Theoretical Physics 21 (1982) 467-488.

[13] L. Grover, A fast quantum mechanical algorithms for datdbase search, in: Proc. 28th Annual ACM Symp. Theory of Computing, Philadelphia, Pennsylvania, 1996, pp. 212-219.

[14] J. Gruska, Quantum Computing, McGraw-Hill, London, 1999.

[15] M. Golovkins, M. Kravtsev, Probabilistic reversible automata and quantum automata, in: Proc. 18th International Computing and Combinatorics Conference

(COCOON'02), Lecture Notes in Computer Science, Vol. 2387, Springer, Berlin, 2002, pp. 574-583.

[16] M. Hirvensalo, Various Aspects of Finite Quantum Automata, in: Proceedings of DLT 2008, Lecture Notes in Computer Science, vol. 5257, Springer, Berlin, 2008, pp. 21-33.

[17] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addision-Wesley, New York, 1979.

[18] E. Jeandel, Topological Automata, Theory Computing Systems 40 (2007) 397-407.

[19] A. Kondacs, J. Watrous, On the power of finite state automata, in: Proceedings of the 38th IEEE Annual Symposium on Foundations of Computer Science, Miami Beach, Florida, USA, 1997, pp. 66-75.

[20] L.Z. Li, D.W. Qiu, Determination of equivalence between quantum sequential machines, Theoretical Computer Science 358 (2006) 65-74.

[21] L.Z. Li, D.W. Qiu, Determining the equivalence for one-way quantum finite automata, Theoretical Computer Science 403 (2008) 42-51.

[22] L.Z. Li, D.W. Qiu, X.F. Zou, L.J. Li, L.H. Wu, Characterizations of one-way general quantum finite automata, arXiv: 0911.3266v1, 2009.

[23] C. Moore, J.P. Crutchfield, Quantum automata and quantum grammars, Theoretical Computer Science 237 (2000) 275-306.

[24] C. Mereghetti, B. Palano, Quantum finite automata with control language, RAIRO-Inf. Theor. Appl. 40 (2006) 315-332.

[25] M.A. Nielsen, I.L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, 2000.

[26] K. Paschen, Quantum finite automata using ancilla qubits, Technical report, University of Karlsruhe, 2000.

[27] A. Paz, Introduction to Probabilistic Automata, Academic Press, New York, 1971.

[28] D. Perrin, Finite automata, In: J. van Leeuwen (Eds.), Handbook of Theoretical Computer Science, Elsevier Science, Holland, 1994, Chap. 1.

[29] D.W. Qiu, L.Z. Li, An overview of quantum computation models: quantum automata, Frontiers of Computer Science in China 2 (2)(2008) 193-207.

[30] D.W. Qiu, S. Yu, Hierarchy and equivalence of multi-letter quantum finite automata, Theoretical Computer Science 410 (2009) 3006-3017.

[31] M. O. Rabin, Probabilistic Automata, Information and Control, 6 (3) (1963) 230-245.

[32] G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 1, Springer-Verlag, Berlin, 1997.

[33] A. Sernadas, P. Mateus, and Y. Omar. Quantum computation and information. In M. S. Pereira, editor, *A Portrait of State-of-the-Art Research at the Technical University of Lisbon*, pages 46–65. Springer, 2007.

[34] P.W. Shor, Algorithm for quantum computation: discrete logarithms and factoring, in: Proc. 37th IEEE Annu. Symp. on Foundations of Computer science, 1994, pp. 124-134.

[35] I. Simon, Piecewise testable events, in: Proc. the 2nd GI conference, Lecture Notes in Computer Science, Vol. 33, Springer, New York, 1975.

[36] W.G. Tzeng, A Polynomial-time Algorithm for the Equivalence of Probabilistic Automata, SIAM Journal on Computing 21 (2) (1992) 216-227.

[37] S. Yu, Regular Languages, In: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Springer-Verlag, Berlin, 1998, pp. 41-110.